สาธิต วงศ์วิวัฒนานนท์

เลือกไฟล์แบบเทอร์โบ

การเขียนโปรแกรมยูทิลิตี้เพื่อจัดเรียงลำดับไดเรคทอรีและชื่อไฟล์เพื่อแสดงให้เลือกได้อย่างรวดเร็ว

ใช้ว่าท่านผู้อ่านที่เคยใช้โปรแกรมประเภทเวิร์ดโปรเซสซิงหรือคอมพายเลอร์ คงจะคุ้นเคยกับการเปิดไฟล์ข้อมูลที่อำนวยความสะดวกในการเลือกไฟล์ ในโปรแกรมสำเร็จรูปที่ดีนั้นจะสามารถแสดงชื่อไฟล์ทั้งหมดขึ้นมาให้เลือกโดยใช้ปุ่มลูกศรและปุ่มฟังก์ชันได้ ยูทิลิตี้ GetFile ที่นำมาเสนอในครั้งนี้จะช่วยให้เขียนโปรแกรมได้โดยง่ายด้วยคำสั่งเพียงบรรทัดเดียว

เมื่อทดสอบยูทิลิตี้ตัวนี้แล้ว ปรากฏว่าโพรซีเดอร์ GetFi สามารถอ่านและจัดเรียงลำดับไดเรคทอรีและชื่อไฟล์เพื่อแสดงให้เลือกได้รวดเร็วมาก เร็วกว่าใน CU-WRITER 1.52 ถึง 3 เท่า ที่ไม่อยากเชื่อคือ เร็วกว่า Qedit เสียอีก ซึ่งจะเห็นได้ชัดเมื่อเลือกจากไดเรคทอรีที่มีไฟล์เป็นร้อยขึ้นไป ที่สามารถทำงานได้รวดเร็วขนาดนี้ ต้องยกให้การเขียนโปรแกรมแบบ procedural type ในโพรซีเดอร์ QuickSort และตัวอย่างโปรแกรม dirdemo.pas ที่มาพร้อมกับเทอร์โบปาสคาลรุ่น 5.5 และ 6.0

ผู้เขียนได้ดัดแปลงโปรแกรม dirdemo.pas ซึ่งแสดงการอ่านและจัดเรียงลำดับไดเรคทอรีและชื่อไฟล์ ต่อเติมในส่วนติดต่อกับผู้ใช้เพื่อให้สามารถใช้ปุ่มลูกศรและปุ่มฟังก์ชันในการเลือกไฟล์ได้ โดยเขียนขึ้นเป็นโพรซีเดอร์ GetFi ซึ่งเป็นโพรซีเดอร์หนึ่งในยูทิลิตี้ GetFile ถ้าจะสังเกตให้ดี ยูทิลิตี้ GetFile นี้มีส่วนคล้ายคลึงกับยูทิลิตี้ Utility ของคุณบุญแสง สุรวัฒนบุตรซึ่งลงพิมพ์ในไมโครคอมพิวเตอร์ฉบับที่ 44 โดยได้เลือกดัดแปลงเอาเฉพาะโพรซีเดอร์ที่จะถูกเรียกใช้โดยโพรซีเดอร์ GetFi และพยายามหลีกเลี่ยงการเขียนโปรแกรมกับฮาร์ดแวร์โดยตรง ทั้งนี้เพื่อให้ง่ายแก่การทำความเข้าใจและไม่ยุ่งยากกับการดัดแปลงโปรแกรมเมื่อใช้ฮาร์ดแวร์ต่างชนิดกัน

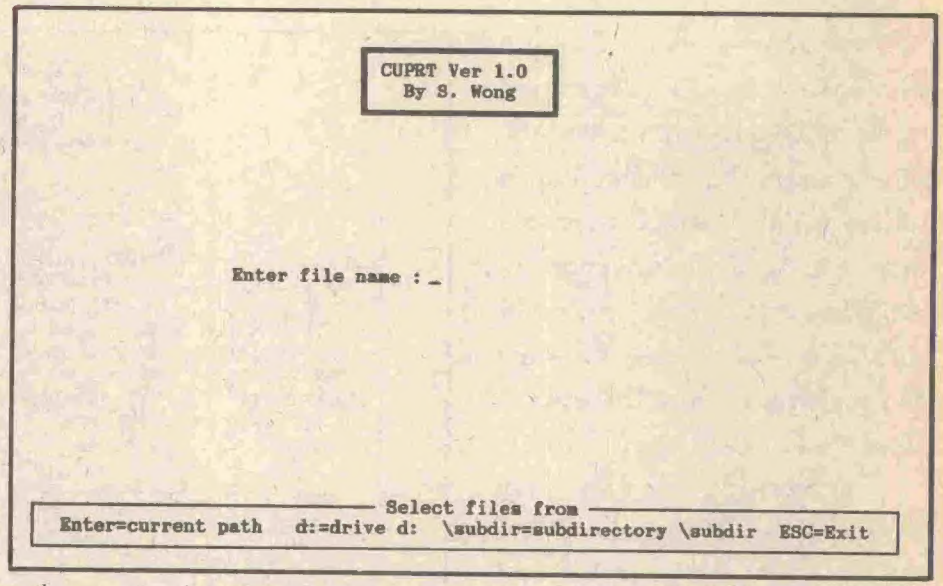
การออกแบบโปรแกรม

ผู้เขียนได้ศึกษาการเลือกไฟล์ด้วยปุ่มลูกศรและปุ่มฟังก์ชันจากหลาย ๆ โปรแกรม สุดท้ายตกลงใจเลือกเอารูปแบบการเลือกไฟล์จากโปรแกรมคอมพายเลอร์เทอร์โบปาสคาล 5.5 และโปรแกรม CU-WRITER 1.52 เป็นหลักในการออกแบบ โดยสรุป ได้ออกแบบให้สามารถใช้งานได้ดังนี้คือ

- เริ่มต้นด้วยการแสดงข้อความและตัวกะพริบให้เติมชื่อไฟล์ โดยให้ผู้ใช้เติมชื่อไฟล์ซึ่งสามารถเติมชื่อพาธตามข้อกำหนดของดอสได้ทุกประการ กรณีไม่ได้ระบุชื่อไฟล์ก็จะแสดงชื่อไฟล์ทั้งหมดหรือตามที่ระบุในแบบ wildcard ให้เลือกต่อไป
- การแสดงชื่อไฟล์ให้เลือกจะแสดงในกรอบสี่เหลี่ยมเรียง 9 แถว ๆ ละ 4 สดมภ์ เรียงตามลำดับชื่อ ต่อด้วยชื่อไดเรคทอรีย่อยเรียงตามลำดับชื่อเช่นกัน
- มีแถบสว่างเพื่อให้ใช้ปุ่มลูกศรทั้งสี่ และปุ่มฟังก์ชันต่าง ๆ ได้ดังนี้

ปุ่มลูกศร	เลือกแถบสว่างไป 1 ไฟล์ตามทิศทางของลูกศร
PgDn	เลื่อนแถบสว่างไปข้างหน้า 1 หน้า กรณีที่อยู่ที่หน้าสุดท้าย จะเลื่อนไปอยู่ไฟล์ล่างสุดในสดมภ์เดียวกัน
PgUp	เลื่อนแถบสว่างถอยหลัง 1 หน้า กรณีที่อยู่ที่หน้าแรกจะเลื่อนไปอยู่ไฟล์บนสุดในสดมภ์เดียวกัน
Home	เลื่อนแถบสว่างไปอยู่ที่ไฟล์แรกสุด
End	เลื่อนแถบสว่างไปอยู่ที่ไฟล์ท้ายสุด

- กดปุ่มตัวอักษรแรกของชื่อไฟล์จะเลื่อนแถบสว่างไปยังไฟล์ที่ขึ้นต้นด้วยอักษรตัวนั้นได้ทันที ถ้ากดปุ่มเดิมซ้ำ จะเลื่อนวนไปในชุดไฟล์ที่ขึ้นต้นด้วยตัวอักษรตัวเดียวกัน
- แสดงไดเรคทอรีแม่ ".." และสับไดเรคทอรีซึ่งนำหน้าด้วยเครื่องหมาย "\" เพื่อเลือกให้แสดงชื่อไฟล์ในไดเรคทอรีนั้น ๆ ได้
- กดปุ่ม Enter เป็นการเลือกไฟล์ที่อยู่ในแถบสว่าง ชื่อไฟล์จะเก็บไว้ในตัวแปร FiGot



รูปที่ 1 ภาพหน้าจอเมื่อเรียกใช้โพรซีเดอร์ GetFi

การใช้งาน

ก่อนจะไปสู่เนื้อหาโปรแกรม เรามาดูผลจากการออกแบบโปรแกรมเสียก่อนว่าลักษณะการใช้งานจะเป็นอย่างไร ทั้งนี้เพื่อให้ทำความเข้าใจเนื้อหาโปรแกรมในตอนท้ายได้ง่ายขึ้น

การใช้งานเมื่อมีการเรียกใช้โพรซีเดอร์ GetFi จะขึ้นข้อความบนหน้าจอดังรูปที่ 1 ขั้นตอนนี้จะให้เติมชื่อไฟล์ตามข้อกำหนดของ DOS อย่างไรก็ได้ เช่น

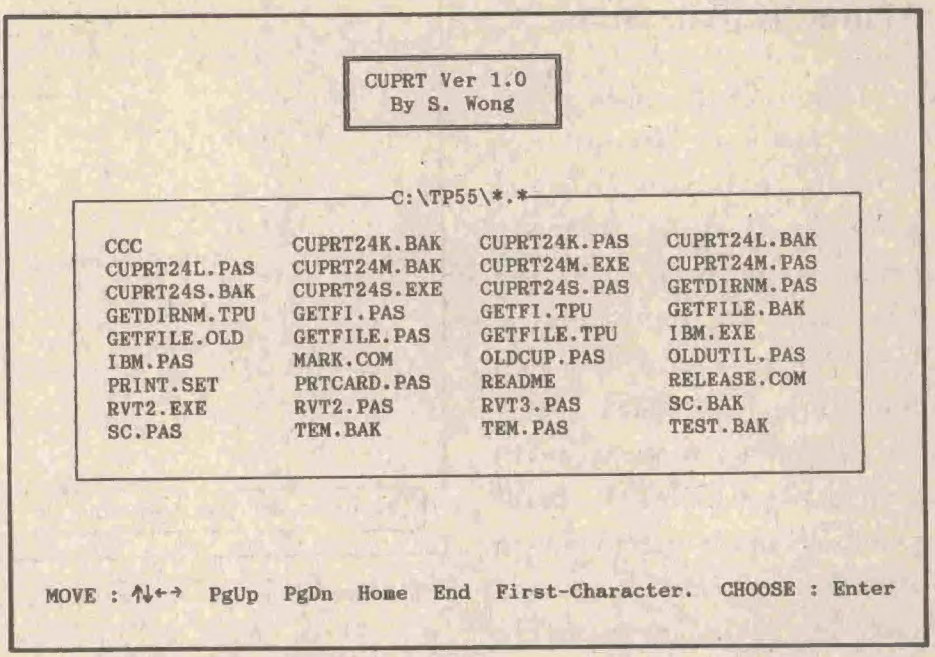
a:	<ชื่อไดรฟ์>
C:\cw156	\ชื่อสับไดเรคทอรี\
c:\cw156\my.doc	<ชื่อไฟล์>
c:\cw156*.doc	<ชื่อไฟล์แบบ wildcard>
<Enter>	<ไม่ใส่ชื่ออะไรเลย = *.* ใน path ปัจจุบัน>

กรณีที่ระบุชื่อไฟล์ด้วย โปรแกรมจะค้นหาชื่อไฟล์นั้น ถ้ามีอยู่จริงก็จะสิ้นสุดโปรแกรม โดยเก็บชื่อไฟล์ไว้ในตัวแปร FiGot เพื่อเรียกใช้ต่อไป กรณีหาชื่อไฟล์ไม่พบหรือกรณีอื่น ๆ จะอ่านไฟล์จากสับไดเรคทอรีหรือไดรฟ์ที่เติมไว้เพื่อแสดงให้เลือกต่อไป

การเติมชื่อไฟล์มี line editor สมบูรณ์แบบ คือสามารถใช้ปุ่มลูกศรซ้าย ขวา ปุ่ม Home End และ BackSpace เพื่อเลื่อนตำแหน่งตัวกะพริบในการแก้ไขได้ นอกจากนี้การกดปุ่ม Esc จะลบข้อความเดิมทิ้งไปและเมื่อไม่ได้เติมอะไร การกด Esc เท่ากับยุติโปรแกรม

การแสดงชื่อไฟล์จะอยู่ในกรอบสี่เหลี่ยมดังที่กล่าวแล้ว ผู้ใช้สามารถกดปุ่ม Home End PgUp PgDn และอักษรแรกของชื่อไฟล์ เพื่อเลื่อนตำแหน่งแถบสว่างไปยังชื่อไฟล์ที่ต้องการ แล้วเคาะ Enter ชื่อไฟล์นั้นก็จะถูกส่งไปเก็บไว้ในตัวแปร FiGot เพื่อเรียกใช้ต่อไปได้ กรณีเคาะ Enter ที่ตำแหน่งไดเรคทอรีก็จะแสดงชื่อไฟล์ในไดเรคทอรีนั้นเพื่อให้เลือกต่อไป (ดูรูปที่ 2)

นอกจากนี้ปุ่ม Esc ยังใช้สำหรับการเปลี่ยนใจออกจากการแสดงชื่อไฟล์กลับไปที่ข้อความแสดงการป้อนชื่อไฟล์ ณ จุดนี้ผู้ใช้สามารถพิมพ์ชื่อไฟล์ตามข้อกำหนดของดอสเพื่อเลือกไฟล์ในไดรฟ์อื่นที่ต้องการได้



รูปที่ 2 ภาพหน้าจอเมื่ออ่านไฟล์ขึ้นจอให้เลือก

อธิบายโปรแกรม

โพรซีเดอร์ GetFi จะทำหน้าที่อ่านชื่อไฟล์ตามข้อกำหนดของดอสโดยอาศัยโพรซีเดอร์ FindFirst และ FindNext ในยูนิตดอสซึ่งชื่อไฟล์ที่อ่านได้จะถูกเก็บไว้ในตัวแปร Dir ซึ่งเป็นอะเรย์ของเรคอร์ดแบบพอยน์เตอร์ จะเห็นว่า Dir ถูกกำหนดไว้ให้เป็นเรคอร์ดขนาด 23 ไบต์ 2 ไบต์แรกเป็นแอตทริบิวต์ 4 ไบต์ถัดมาเป็นเวลาในรูปของ LongInt อีก 4 ไบต์ถัดมาเป็นขนาดไฟล์ในรูปของ LongInt เช่นกัน 13 ไบต์สุดท้ายเป็นชื่อไฟล์ซึ่งได้เผื่อไว้ 1 ไบต์สำหรับเพิ่มเครื่องหมาย "\" กรณีที่เป็นชื่อของสับไดเรคทอรี

```
program CuPrt;
uses crt,getFile;
begin
    clrScr;
    WriteStr(32,2,reverse,' CUPRT Ver 1.0  ');
    WriteStr(32,3,reverse,'   By S. Wong   ');
    WriteBox(31,1,48,4,reverse,box2);
    getFi;
    clearWin(1,6,80,25);
    writeStr(20,10,low,'You have got file '+ fiGot);
    writeStr(20,12,low,'The file will be printed out in');
    writeStr(20,13,low,'the  manner  that  CUPRINT can not do. It');
    writeStr(20,14,low,'will  print up to 45 lines in  text  mode');
    writeStr(20,15,low,'with 13 inches paper at faster speed than');
    writeStr(20,16,low,'CUPRINT.');
    writeStr(35,17,high+blink,'WAIT AND SEE');
    cursorOff;
end.
```

รูปที่ 3 ตัวอย่างโปรแกรมเรียกใช้ยูนิต GetFile

จะเห็นว่าที่ต้นยูนิต GetFile นี้กำหนดให้ขนาดฮีฟสูงสุดและต่ำสุดเป็น 7168 ไบต์ (โดยไดเรคตีฟ $M) ทั้งนี้ประมาณค่าได้จากจำนวนไฟล์สูงสุดที่จะกำหนดให้อ่านได้คูณกับขนาดของเรคอร์ด (คือ 23 ไบต์) นั่นเอง ท่านผู้อ่านสามารถแก้จำนวนไฟล์สูงสุดที่ต้องการได้ที่ค่าคงที่ MaxDirSize พร้อมทั้งแก้ขนาดฮีฟด้วย ก่อนที่จะเปิดเนื้อที่หน่วยความจำด้วย GetMem ได้ทำการ mark ตำแหน่งเริ่มต้นของฮีฟไว้ก่อนเพื่อ release หน่วยความจำเดิมออกเมื่อมีการอ่านไดเรคทอรีใหม่หรือเมื่อจบโพรซีเดอร์ ทั้งนี้เพื่อให้ใช้หน่วยจำเท่าที่กำหนดให้อย่างประหยัดที่สุดในการเลือกเปิดไฟล์

โพรซีเดอร์ GetFi จะเริ่มทำงานโดยการเรียกใช้โพรซีเดอร์ย่อย ๆ ตามลำดับดังนี้

โพรซีเดอร์ GetCommand จะเคลียร์พื้นที่หน้าจอตั้งแต่บรรทัดที่ 6 ไปจนบรรทัดล่างสุด แล้วแสดงข้อความ Enter file name และ Help menu ที่บรรทัดล่างสุด พื้นที่จอ 5 บรรทัดแรกเหลือไว้สำหรับผู้เขียนโปรแกรมเขียนหัวเรื่องหรือข้อความที่ต้องการเอาเองในโปรแกรมของตน (ดูตัวอย่างการเรียกใช้โปรแกรมในตอนท้าย) GetCommand จะรับสตริงที่พิมพ์เข้าไป จากนั้นจะทำการตรวจสอบว่าเป็นชื่อไฟล์หรือไดเรคทอรี และมีอยู่จริงหรือไม่ตามที่กล่าวแล้ว ถ้าเป็นชื่อไฟล์ที่มีอยู่จริงก็จะจบโปรแกรมที่นี่เลย

โพรซีเดอร์ FindFiles จะทำการอ่านชื่อไฟล์ตามที่ต้องการด้วยโพรซีเดอร์ FindFirst และ FindNext เก็บไว้ในตัวแปร Dir ตามที่กล่าวแล้ว

โพรซีเดอร์ DefindDirectory ตรวจสอบชื่อที่อ่านได้ทั้งหมด ถ้าชื่อใดเป็นไดเรคทอรี (คือมีแอตทริบิวต์เท่ากับ 16) ก็จะเติม '\' ไว้หน้าชื่อ

โพรซีเดอร์ SortFiles จัดเรียงลำดับชื่อไฟล์และไดเรคทอรีด้วยความเร็วแบบเทอร์โบ

โพรซีเดอร์ PrintFiles เป็นส่วนติดต่อกับผู้ใช้ โดยแสดงชื่อไฟล์ในกรอบสี่เหลี่ยมและ Select Menu ที่บรรทัดล่างสุด มีโพรซีเดอร์ย่อยสำหรับให้ผู้ใช้กดปุ่มลูกศรและปุ่มฟังก์ชันเลือกไฟล์ตามที่กล่าวแล้ว

นอกจากนี้ในยูนิต GetFile ยังมีโพรซีเดอร์อื่น ๆ ซึ่งท่านผู้อ่านสามารถเรียกใช้ได้ โพรซีเดอร์เหล่านี้ถูกเรียกใช้โดยโพรซีเดอร์ GetFi เช่นกัน ได้แก่

โพรซีเดอร์ CursorOn และ CursorOff สำหรับสั่งแสดงและไม่แสดงตัวกะพริบ

โพรซีเดอร์ ClearWin(x1,y1,x2,y2:byte) สำหรับเคลียร์พื้นที่จอตามกรอบที่ระบุ x1,y1 เป็นตำแหน่งมุมซ้ายบน และ x2,y2 เป็นตำแหน่งมุมขวาล่างของกรอบ

โพรซีเดอร์ writeStr(x,y,attr:byte;st:string) แสดงข้อความ st บนจอภาพที่ตำแหน่ง x,y ตามแบบตัวอักษร attr ตามข้อกำหนดของ DOS (ระบุไว้ในค่าคงที่แล้ว เช่น low=7, high=15 .....)

โพรซีเดอร์ writeBox(x1,y1,x2,y2,attr:integer;BoxData:BoxType) ตีกรอบตารางที่ตำแหน่งมุมซ้ายบน(x1,y1) และมุมขวาล่าง(x2,y2) ตามแบบตัวอักษร attr ที่กำหนด กรอบเส้นเดี่ยวหรือคู่เป็นไปตามแบบ BoxType ในค่า

โพรซีเดอร์ ReadStr (F:PathStr) สำหรับป้อนข้อความจากแป้นพิมพ์เก็บไว้ใน F โดยมี line editor สมบูรณ์แบบและมีปุ่ม ESC เพื่อยกเลิกข้อความหรือออกจากโพรซีเดอร์

ทดลองเรียกใช้ยูนิต GetFile

โปรแกรมที่เขียนขึ้นเพื่อเรียกใช้ยูนิต GetFile หากมีการกำหนดค่าสแต็กฮีฟสูงสุดและต่ำสุดจะต้องมีค่าเท่ากับที่กำหนดไว้ในยูนิตนี้เสมอ ตัวอย่างดูได้จากรูปที่ 3

สรุป

ยูนิต GetFile เขียนขึ้นเพื่อให้โปรแกรมเมอร์เรียกใช้ในส่วนของการเลือกไฟล์ได้ด้วยคำสั่งเดียวคือ GetFi ซึ่งสามารถอ่านและเรียงลำดับไฟล์เพื่อให้เลือกได้รวดเร็วมาก นอกจากนี้ยังมีโพรซีเดอร์พื้นฐานอื่น ๆ ที่ทำให้การเขียนโปรแกรมด้วยเทอร์โบปาสคาลเป็นเรื่องไม่ยากเย็นอะไรเลย เพื่อให้ยูนิตนี้สมบูรณ์ขึ้น ท่านผู้อ่านอาจเขียนโพรซีเดอร์ ReadInteger เพิ่มเติมเองได้

รูปที่ 4
โปรแกรมยูนิต
GetFile

```
unit GetFile;
{$K 7168,7168,7168}

interface

uses Crt, DOS;

const MaxDirSize = 310;

type
  DirPtr   = ^DirRec;
  DirRec   = Record
               Attr: Byte;
               Time: Longint;
               Size: Longint;
               Name: string[13];
             end;
  DirList  = Array[0..MaxDirSize - 1] of DirPtr;
  Boxtype  = Array[1..8] of byte;
```

รูปที่ 4 (ต่อ)

```
const
  Home=#71;     Kend=#79;    PgUp=#73;     PgDn=#81;
  UArr=#72;     DArr=#80;    RArr=#77;     LArr=#75;
  Return=#13;   ESC=#27;     BkSp=#8;      Del =#83;

  box0  : boxtype = ( 32, 32, 32, 32, 32, 32, 32, 32);
  box1  : boxtype = (218, 196, 191,179,217,196,192,179);
  box2  : boxtype = ( 201,205,187,186,188,205,200,186);
  box3  : boxtype = (213,205,184,179,190,205,212,179);
  box4  : boxtype = (214,196,183,186,189,196,211,186);

  low  = 7;     high  = 15;    reverse  = 112;
  low_ = 1;     high_ =  9;      blink  = 128;


var
  Count,cur: Integer;
  FiGot : PathStr;
  Path: PathStr;
  Dir: DirList;
  htop : ^integer;
  ch : char;
  D: DirStr;
  N: NameStr;
  E: ExtStr;
  ReStart, FindPath : Boolean;

procedure CursorOn;
procedure CursorOff;
procedure ClearWin(x1,y1,x2,y2:byte);
procedure WriteStr(x,y,attr : byte; st : string);
procedure WriteBox(x1,y1,x2,y2,attr:integer;BoxData:BoxType);
procedure ReadStr(var F:PathStr);
procedure GetFi;

implementation

procedure CursorOn;
var   regs   : Registers;
begin
   regs.ah:=$01;
   regs.ch:=11;
   regs.cl:=12;
   intr($10,regs);
end;

procedure CursorOff;
var   regs   : Registers;
begin
   regs.ah:=$01;
   regs.ch:=14;
   regs.cl:=0;
   Intr($10,regs);
end;


procedure ClearWin(x1,y1,x2,y2:byte);
begin
   Window(x1,y1,x2,y2);ClrScr;Window(1,1,80,25);
end;

procedure WriteStr(x,y,attr : byte; st : string);
begin
    Window(1,1,80,25);
    GotoXY(x,y);
    TextAttr:=attr;
    Write(st);
    TextAttr:=low;
end;

procedure WriteBox(x1,y1,x2,y2,attr:integer;boxdata:boxtype);
var  loop : integer;
begin
   WriteStr(x1,y1,attr,chr(boxdata[1]));
   WriteStr(x2,y1,attr,chr(boxdata[3]));
   WriteStr(x2,y2,attr,chr(boxdata[5]));
   WriteStr(x1,y2,attr,chr(boxdata[7]));
   for loop:=y1+1 to y2-1 do
   begin
      WriteStr(x1,loop,attr,chr(boxdata[8]));
      WriteStr(x2,loop,attr,chr(boxdata[4]));
   end;
   for loop:=x1+1 to x2-1 do
   begin
      WriteStr(loop,y1,attr,chr(boxdata[2]));
      WriteStr(loop,y2,attr,chr(boxdata[6]));
   end;
end;

procedure ReadStr(var F:PathStr);
var i,x,OrgX:byte;
begin
   F:='';x:=whereX;OrgX:=whereX;
   repeat
    GotoXY(x,whereY);
    ch:=ReadKey;
    case ch of
      #0       : begin
                   ch:=ReadKey;                [ Function Key ]
                   case ch of
                    Home : x:=OrgX;
                    Kend : x:=OrgX+length(F);
                    LArr : if x > OrgX then dec(x);
                    RArr : if x < OrgX + length(F) then inc(x);
                    Del  : begin
                             Delete(F,x-OrgX+1,1);
                             GotoXY(OrgX,whereY);ClrEol;write(F);
                           end;
                   end;
                  end;
      BkSp     : if x>OrgX then
                  begin
                   Delete(F,x-OrgX,1);
                   GotoXY(OrgX,whereY);ClrEol;write(F);
                   dec(x);
                  end;
      ESC      : if F='' then F:=#27 else
                   begin
                    for i:=1 to Length(F) do Write(#8); ClrEol;
                    F:='';x:=OrgX;
                   end;
```

รูปที่ 4 (ต่อ)

```
      ' '..'z' : begin
                   Insert(ch,F,x-OrgX+1);
                   inc(x);
                   GotoXY(OrgX,whereY);ClrEol;write(F);
                 end;
    end;
   until (ch=#13) or (F=#27);
end;

{$F+}

function Less(X, Y: DirPtr): Boolean;
begin
  Less:= X^.Name < Y^.Name;
end;
{$F-}

procedure QuickSort(L, R: Integer);
var
  I, J: Integer;
  X, Y: DirPtr;
begin
  I := L;
  J := R;
  X := Dir[(L + R) div 2];
  repeat
    while Less(Dir[I], X) do Inc(I);
    while Less(X, Dir[J]) do Dec(J);
    if I <= J then
    begin
      Y := Dir[I];
      Dir[I] := Dir[J];
      Dir[J] := Y;
      Inc(I);
      Dec(J);
    end;
  until I > J;
  if L < J then QuickSort(L, J);
  if I < R then QuickSort(I, R);
end;

procedure HelpMenu;
begin
 WriteBox(2,23,79,25,low,box1);
 WriteStr(33,23,low,' Select files from ');
 WriteStr(10,24,low,
   '=current path  =drive d:  =subdirectory \subdir =Exit');
 WriteStr(5,24,high,'Enter'); WriteStr(26,24,high,'d:');
 WriteStr(39,24,high,'\subdir'); WriteStr(69,24,high,'ESC');
end;

procedure GetCommand;
var
  I,J: Integer;
  Attr: Word;
  F: File;
begin
  CursorOn;
  ClearWin(1,6,80,25);
  FindPath:=True;
  HelpMenu;
  WriteStr(20,12,low,'Enter file name : ');
  ReadStr(Path);CursorOff;
  if Path=#27 then begin FiGot:='';Exit;end;
  if Path='' then GetDir(0,Path);
  Path := FExpand(Path);
  if Path[Length(Path)] <> '\' then
  begin
    Assign(F, Path);
    GetFAttr(F, Attr);
    if (DosError = 0) then
     begin
      if (Attr and Directory <> 0) then
       Path := Path + '\'
      else
       begin FiGot:=path; FindPath:=False; ReStart:=False; end;
     end;
  end;
  FSplit(Path, D, N, E);
  if (N = '') and (E = '') then
   begin
     N := '*';
     E := '.*';
   end;
  Path := D + N + E;
end;

procedure FindFiles;
var
  F: SearchRec;
begin
  FindPath:=True;
  mark(htop);
  Count := 0;
  FindFirst(Path, ReadOnly + Directory + Archive, F);
  if DosError <> 0 then
    begin
     WriteStr(12,6,low,'...File Not Found...');
     N:='*'; E:='.*'; path:=D +N +E;
     FindFirst(Path, ReadOnly + Directory + Archive, F);
    end;
  If F.Name='.' then  FindNext(F);
  while (DosError = 0) and (Count < MaxDirSize) do
  begin
    GetMem(Dir[Count], Length(F.Name) + 11);
    Move(F.Attr, Dir[Count]^, Length(F.Name) + 10);
    Inc(Count);
    FindNext(F);
  end;
  if N +E <> '*.*' then
    begin
     FSplit(path, D, N, E);
     FindFirst(D+'*.*',ReadOnly +Directory +Archive, F);
     If F.Name='.' then  FindNext(F);
     while (DosError = 0) and (Count < MaxDirSize) do
      begin
       if F.attr = 16 then
       begin
```

รูปที่ 4 (ต่อ)

```
        GetMem(Dir[Count], Length(F.Name) + 11);
        Move(F.Attr, Dir[Count]^, Length(F.Name) + 10);
        Inc(Count);
       end;
       FindNext(F);
      end;
    end;
end;

procedure DefineDirectory;
var i:integer;
begin
     for i:=0 to count-1 do
         with Dir[i]^ do
           if Attr =16 then Name:='\' + Name;
end;

procedure SortFiles;
begin
  if (Count <> 0) and (@Less <> nil) then
    QuickSort(0, Count - 1);
end;

procedure PrintFiles;
var
  x,y,i:byte;
  dispmin,dispmax : integer;

  procedure ListFile;
  var i : integer;
  begin
      Window(11,9,70,18);ClrScr;
      LowVideo;
      for i:= dispmin to dispmax do
      with Dir[i]^ do Write(' ',Name,' ':14-length(Name));
  end;

  procedure Right;
  begin
            if cur < count-1 then Inc(cur) else Exit;
            if cur <= dispmax then
             begin
             if cur MOD 4 <> 0 then Inc(x) else
             begin Inc(y);x:=0;end;
            end
           else
            begin
            Inc(dispmin,4); Inc(dispmax,4);
            if dispmax > count-1 then dispmax:=count-1;
            ListFile;
            x:=0;
            end;
  end;

  procedure Left;
  begin
    if cur > 0 then Dec(cur) else Exit;
    if cur >=dispmin then
     begin
      if cur MOD 4 <> 3 then Dec(x) else
       begin Dec(y); x:=3; end;
     end
    else
     begin
      Dec(dispmin,4);
      if dispmax - dispmin > 35 then dispmax:=dispmin + 35;
      x:=3;
      ListFile;
     end;
end;

procedure Up;
begin
  if cur > 3 then Dec(cur,4) else Exit;
  if cur >= dispmin then Dec(y) else
   begin
    Dec(dispmin,4);
    if dispmax - dispmin > 35 then dispmax:=dispmin + 35;
    ListFile;
   end;
end;

procedure Down;
begin
  if cur < count-4 then Inc(cur,4) else
   begin
   if dispmax < count-1 then
    begin
     x:=(count-1) MOD 4;
     if y < 8 then Inc(y);
     Inc(dispmin,4);
     dispmax:=count-1;
     cur:=count-1;
     ListFile;
     Exit;
    end
   else Exit;
   end;
  if (cur <= dispmax) then Inc(y) else
   begin
    Inc(dispmin,4); Inc(dispmax,4);
    if dispmax > count-1 then dispmax:=count-1;
    ListFile;
   end;
end;

procedure PageUp;
begin
  if dispmin=0 then begin y:=0; cur:=x; Exit; end;
  if dispmin > 35 then
    begin
     Dec(cur,36);
     Dec(dispmin,36);
     dispmax:=dispmin+35;
    end
  else
    begin
     dispmin:=0;
```

รูปที่ 4 (ต่อ)

```
      dispmax:=35;
      cur:=x+y*4;
     end;
   ListFile;
  end;

  Procedure PageDown;
  begin
   if dispmax=count-1 then
    begin
     y:=(dispmax-dispmin) DIV 4;
     if X > (dispmax-dispmin) MOD 4 then Dec(y);
     cur:=y*4+x+dispmin;
     Exit;
    end;
   if dispmax < count - 36 then
    begin
     Inc(cur,36);
     Inc(dispmin,36);
     Inc(dispmax,36);
    end
   else
    begin
     dispmax:=count-1;
     dispmin:=count-33-((count-1) MOD 4 );
     cur:=y*4+x+dispmin;
     if y=8 then if (count-1) MOD 4 < x then
      begin
       x:=(dispmax-dispmin) MOD 4;
       cur:=count-1;
      end;
    end;
   ListFile;
 end;

 procedure GoHome;
 begin
  x:=0; y:=0; cur:=0;
  if dispmin=0 then Exit;
  dispmin:=0;
  if count<37 then dispmax:=count-1 else dispmax:=35;
  ListFile;
 end;

 procedure GoEnd;
 begin
  cur:=count-1;
  if dispmax=count-1 then
   begin
    x:=(dispmax-dispmin) MOD 4;
    y:=(dispmax-dispmin) DIV 4;
   end
  else
   begin
    dispmax:=count-1;
    dispmin:=count-33-((count-1) MOD 4 );
    y:=8; x:=(dispmax-dispmin) MOD 4;
    ListFile;
   end;
 end;

  procedure SearchCh;
  var i:integer;c:char;
  begin
   c:=Upcase(ch);
   if (c=Dir[cur+1]^.Name[1]) and (cur+1 < count) then
    Inc(cur)
   else
    begin
     i:=-1;
     repeat
      Inc(i);
     until (c=Dir[i]^.Name[1]) or (i=count);
     if i <> count then cur:=i;
    end;

  end;

  procedure FindChar;
  begin
   SearchCh;
   if cur in [dispmin..dispmax] then
    begin
     x:=cur MOD 4;
     y:=(cur-dispmin) DIV 4;
    end
   else
    begin
     x:=cur MOD 4;
     dispmin:=cur-x;
     dispmax:=dispmin+35;
     if dispmax > count-1 then dispmax:=count-1;
     y:=(cur-dispmin) DIV 4;
     ListFile;
    end;
  end;

  procedure SelectMenu;
  var i:integer;
  begin
   WriteStr(1,24,low,'      MOVE : ');
   WriteStr(14,24,high,chr(24)+chr(25)+chr(26)+chr(27)
    +' PgUp PgDn Home End First-Character.');
   WriteStr(70,24,high,'Enter      ');
   WriteStr(59,24,low,'  CHOOSE : ');
  end;

begin  { PrintFile }
 CursorOff;
 ClearWin(1,12,80,25);
 SelectMenu;
 WriteBox(9,7,72,19,high,box1);
 WriteStr(40-(length(path) DIV 2),7,high,path);
 Window(11,9,70,18);
 if Count = 0 then
  begin
   WriteLn('No matching files. Any key to continue...');
   FindPath:=False;ReStart:=True;
   ch:=readkey;
   Exit;
  end
```

รูปที่ 4 (ต่อ)

```
else ReStart:=False;
x:=0;y:=0;dispmin:=0;cur:=0;
if count<37 then dispmax:=count-1 else dispmax:=35;
ListFile;
repeat
  with Dir[cur]^ do
  begin
   WriteStr(x*15+11,y+9,reverse,' '+Name+' ');
   ch:=readkey;
   WriteStr(x*15+11,y+9,low,' '+Name+' ');
  end;
  if ch=#0 then
   begin
    ch:=readkey;
    case ch of
     RArr : Right;
     LArr : Left;
     UArr : Up;
     DArr : Down;
     PgUp : PageUp;
     PgDn : PageDown;
     Home : GoHome;
     KEnd : GoEnd;
    end;
   end
  else if ((ch<>Return) or (ch<>ESC)) then FindChar;
until ch in [Return,ESC];
if ch=ESC then
 begin
  ReStart:=True; FindPath:=False;

   end
  else
   begin  {ch=Return}
    FSplit(path, D, N, E);
    with Dir[cur]^ do
     if Name[1]='\' then
       begin
        if Name <> '\..' then
         begin
         delete(Name,1,1);
         path:=D + Name + '\'+ N + E;
         end
        else
         begin
          i:=length(D);
          repeat
           Dec(i);
          until D[i]='\';
          D:=copy(D,1,i);
          path:=D +N +E;
         end;
       end
      else
       begin
        PiGot:=D +Dir[cur]^.Name;
        FindPath:=False;
       end;
     ReStart:=False;
    end;   {ch=Retuen}
    release(htop);
end; { PrintFile }

procedure GetFi;
begin
 repeat
   GetCommand;if Path=ESC then Exit;
   while FindPath do
    begin
     FindFiles;
     DefineDirectory;
     SortFiles;
     PrintFiles;
    end;
 until not ReStart;
 Window(1,1,80,25);CursorOn;
end;

END.
```

■■■